Electronic Version

Stylesheet Version v1.1.1

Description

METHOD OF SECURE DATA EXCHANGE

BACKGROUND OF THE INVENTION

[0001] The present invention relates to a method of secure data exchange, and more particularly to an encryption key exchange between two cryptographic units.

[0002] Two mutually exclusive classes of cryptographic methods and protocols are well known to those familiar with cryptography, symmetric cryptography, and public-key cryptography (or named asymmetric cryptography). In symmetric cryptographic protocols, the same key and cryptographic method are used both for encrypting a plaintext into cyphertext and for decrypting a cyphertext to recover the plaintext. It is readily apparent that the security of a symmetric cryptographic protocol can never exceed the security of the single key used for both encryption and decryption.

[0003] For symmetric cryptographic protocols, there are three well-known key management problems. First, a key may be compromised, which permits an eavesdropper who obtains the key either to read all the cyphertext or even to broadcast bogus cyphertext. The only way to alleviate this problem is to change the key frequently. A second problem for symmetric cryptography key management is that it requires a large number of keys if each pair of individuals in a group is to communicate with each other using a different key. Forty-five unique keys are required if a group of 10 individuals are to communicate. Fifty-five unique keys are required for communication among a group of 11 individuals. The final problem for key management in symmetric cryptographic protocols is that, since the keys are more valuable than the encrypted message, the keys must be exchanged by a secure communication.

[0004] Whether used with a symmetric cryptographic protocol or with a public-key cryptographic protocol, an encryption key should not be used indefinitely. First, the longer time a key is used for, the more likely it will be comprised by theft, luck, extortion, bribery or cryptanalysis. Longer use of a key aids an eavesdropper because it provides more cyphertext encoded with the same key to which cryptanalytic methods may be applied. Second, usually the longer time a key is used for, the greater loss the key must compromise on.

SUMMARY OF THE INVENTION

[0005] The primary objective of the present invention is to provide a method of secure data exchange, which is secure against cryptanalysis. The method is based on the exchange of cryptographic keys between two cryptographic units. Because a new key replaces a previous key during every session, an eavesdropper steals too little cyphertext to complete cryptanalysis.

[0006] Another objective of the present invention is to provide a cryptographic key exchange protocol that is simpler than the present protocols.

[0007] In order to achieve the objective, the present invention discloses a method of secure data exchange occurring in a system that includes a server and at least a client. An initial key is either pre-configured by factories and permanently stored in the client (or named as an endpoint) or obtained from the server through a manual login. After starting to first connect to the server, the client sends a reset message to the sever using the initial key. Once receiving the message, the server verifies the received messages and also uses the initial key to decrypt them. If the verification of the messages is approved, the server generates a first key and sends a key exchange message that includes the first key and is encrypted by the initial key to the client. Afterward, the client replaces the initial key with the first key in response to the received key exchange message, and meanwhile returns a key confirmation message. If the key confirmation message is approved, the server sends a downloading message to allow the client to retrieve corresponding information. After the information is successfully downloaded, the client sends a finish message to notify the server to await the coming of a next session.

[0008] When the next session starts, the client sends a key validation message encrypted by the first key. If the key validation message is approved, the server grants the request of the client to send another key exchange message with a second key. Conversely, the initial key is still used to request the approval of the server.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The invention will be described according to the appended drawings in which:

[0010] FIG. 1 is a diagram of message transition between two cryptographic units in accordance with the preferred embodiment of the present invention;

[0011] FIG. 2 is a diagram of message transition between two cryptographic units in accordance with the preferred embodiment of the present invention;

[0012] FIG. 3 is a diagram of message transition between two cryptographic units in accordance with the preferred embodiment of the present invention;

[0013] FIG. 4 is a diagram of message transition between two cryptographic units in accordance with the preferred embodiment of the present invention;

[0014] FIG. 5 is a diagram of message transition between two cryptographic units in accordance with the preferred embodiment of the present invention; and

[0015] FIG. 6 is a diagram of state transitions regarding the server in accordance with the present invention.

PREFERRED EMBODIMENT OF THE PRESENT INVENTION

[0016] In an automatic provisioning system (APS), clients can obtain all configuration data from an automatic provisioning server, and all communication including the configuration data between them is encrypted and secure against any eavesdropper. Two entities are involved in the system, one being a key distribution server (KDS) which holds all client's profiles and cryptographic information, such as key information, and the other being the endpoints (EPs) maintained by the clients. Furthermore, the KDS can be integrated into the automatic provisioning server.

[0017] The KDS, acting as a powerful server, holds clients' personal data and execute complicated encryption and decryption processes. Generally speaking, the computational capability and storage capacity of EPs are limited by themselves. Therefore, the present invention discloses that a method improves the security of data exchange between the two entities regardless of the enhancement of the EPs' performance.

[0018] The following notations are used to explain the key distribution mechanism of a KDS in response to the requirement of an EP:

[0019] S(i) denotes the states of the i-th endpoint (EPi) monitored by the KDS;

[0020] K(i) denotes the key held by the EPi;

[0021] RK(i) denotes an initial key stored in the EPi;

[0022] AK(i) denotes a current active key held by the KDS;

[0023] CK(i) denotes a next key designated by the KDS; and

[0024] MAC(i) denotes the MAC (Media Access Control) address of the EPi.

[0025] There are four states of the EPi monitored by the KDS. First, an inactive state means the EPi is not recognized by the KDS; an exchanging state means a key exchange occurs between them; an idle state means the EPi is recognized by the KDS and awaits the coming of a next event initialized by either itself or the KDS; finally, a downloading state means the KDS allows the EPi to download data from somewhere that is authorized. In summary, the definitions of all messages occurring between the EPs and KDS are summarized in Table 1.

[0026]

Table 1: Definition of Messages		
Name	Initiator	Description
RESET	EP	Reset a current key for the EP.
KEY_EXCHANGE	KDS	Distribute a new key.
KEY_CONFIRM	EP	Confirm if a current key is received.
KEY_VALIDATE	EP	Validate a current key.
EP_INACTIVE	KDS	Indicate if the EP is inactive.
EP_INVALID	KDS	Invalidate a current key.
EP_INFO	KDS	Send related downloading message.
COMPLETE	EP	Finish downloading.

[0027] FIG. 1 is a diagram of message transitions between two cryptographic units in accordance with the preferred embodiment of the present invention. An initial key RK(i) is either pre-configured by factories and permanently stored in the EPi or obtained from the KDS through a manual login. After starting to first connect to the KDS, the EPi sends a reset message RESET, encrypted by the initial key RK(i), to the KDS through broadcast. Once receiving the reset message including the physical address MAC(i) of the EPi, a timestamp of 4 bytes and a one-way hash value, the KDS verifies the received message and also uses the initial key RK(i) to decrypt them. If the verification of the messages is approved, the KDS generates a first key CK(i) and sends a key exchange message KEY_EXCHANGE, which includes the first key CK(i) and is encrypted by the initial key RK(i) to the EPi. Afterward, the EPi replaces the initial key RK(i) with the first key CK(i) in response to the received key exchange message, and meanwhile returns a key confirmation message KEY_CONFIRM. If the key confirmation message is approved, the KDS sends a related downloading message of EP_INFO to allow the EPi retrieving the corresponding information from a destination address and a path specified by the downloading message. After all information is successfully downloaded, the EPi sends a finish message COMPLETE to notify the server to await the coming of a next session.

[0028] When the next session starts, the client sends a key validation message KEY_VALIDATE, encrypted by the first key CK(i). If the key validation message is approved, the KDS grants the request of the EPi to send another key exchange message with a second key in place of the first key.

[0029] After the communication between the EPi and KDS continues for several sessions, the EPi sends a key validation message encrypted by the current key K(i) to the KDS, as shown in FIG. 2. Unfortunately, if the validation message is disapproved by the KDS because of the errors existing in its certain packets, the KDS returns a key invalidation message EP_INVALID. Afterward, the EPi continuously sends a reset message of RESET encrypted by the initial key RK(i) to access the KDS. Referring to FIG. 2, the succeeding message transitions from KEY_EXCHANGE to COMPLETE are the same as the corresponding transitions in FIG. 1.

[0030] Please refer to FIG. 3, in comparison with the case in FIG. 2, after sending a reset message RESET, encrypted by the initial key RK(i), to access the KDS, the EPi is not recognized and is notified with an inactive EP message EP_INVALID by the KDS.

[0031] Referring to FIG. 4, if the EPi sends a reset message RESET, encrypted by the initial key RK(i), to access the KDS. Unfortunately, the EPi is not recognized and is notified with an inactive EP message EP_INVALID by the KDS.

[0032] As shown in the last case of FIG. 5, after some sessions, the EPi sends a key validation message KEY_VALIDATE, encrypted by the initial key RK(i), to the KDS, but in vain. The EPi is not recognized and is also notified with an inactive EP message EP_INVALID by the KDS.

[0033] The packet structures of all aforesaid messages are summarized in Table 2. Each packet data includes a message ID, field 1 and field 2. All main data including cryptographic keys, address information, timestamp data, etc., are stored in Field 1, and one-way hash values are stored in Field 2.

[0034]

Table 2: Packet Structure of Messages		
Message ID	Field 1	Field 2
RESET	E(RK(i), MAC(i)(6 bytes) +timestamp(4 bytes))	HASH(RK(i))
KEY_EXCHANGE	E(AK(i), CK(i))	HASH(AK(i))
KEY_CONFIRM	E(CK(i), MAC(i)(6 bytes) +timestamp(4 bytes))	HASH(CK(i))
KEY_VALIDATE	E(K(i), MAC(i)(6 bytes)	HASH(K(i))

	+timestamp(4 bytes))	
EP_INACTIVE	N/A	
EP_INVALID	N/A	
EP_INFO	E(CK(i), Address(4 bytes)+ Path (32 bytes))	HASH(CK(i))
COMPLETE	E(K(i), MAC(i)(6 bytes) +timestamp(4 bytes))	HASH(K(i))

[0035] FIG. 6 is a diagram of state transitions regarding the server in accordance with the present invention. Four states of the EPi are monitored by the KDS, which are in-active state 61, idle state 62, exchange state 63 and downloading state 64. When the EPi is at the idle state, it can decide to communicate with the KDS by means of the transmission of the reset message or the key validation message. However, the EPi is considered as an inactive endpoint after all efforts fail to access the KDS. As shown in Step 65, if the KDS verifies the transmitted messages from the EPi, the EPi is changed into the exchange state; otherwise, the EPi receives a disapproval message EP_INVALID from the KDS. When the EPi is at the exchange state, it can send the reset message or the key validation message to the KDS to restore the communication with the KDS. After the key exchange is done, the KDS needs to check whether the EPi correctly receives the new key within the Step 66. If the key confirmation fails to be approved, then the exchange state will still appear. Otherwise, the downloading state is a coming state. After all data are downloaded from a destination, such as an auto-provisioning server, the downloading state is changed into the idle state. The aforesaid transition of states and messages are summarized in Table 3.

[0036]

Table 3: State Transition Table for KDS				
STATE / MESSAGE	INACTIVE	EXCHANGING	DOWNLOADING	IDLE
RESET (Correct)	INACTIVE / EP_INACTIVE	EXCHANGING / KEY_EXCHANGE	EXCHANGING / KEY_EXCHANGE	EXCHANGING / KEY_EXCHANGE
KEY_CONFIRM (Correct)	INACTIVE / EP_INACTIVE	DOWNLOADING/ EP_INFO	DOWNLOADING/ EP_INFO	IDLE /
KEY_VALIDATE (Correct)	INACTIVE / EP_INACTIVE	EXCHANGING / KEY_EXCHANGE	EXCHANGING / KEY_EXCHANGE	EXCHANGING / KEY_EXCHANGE
COMPLETE (Correct)	INACTIVE / EP_INACTIVE	IDLE /	IDLE /	IDLE /
RESET (Incorrect)	INACTIVE / EP_INACTIVE	EXCHANGING / EP_INVALID	DOWNLOADING/ EP_INVALID	IDLE / EP_INVALID
KEY_CONFIRM (Incorrect)	INACTIVE / EP_INACTIVE	EXCHANGING / EP_INVALID	DOWNLOADING/ EP_INVALID	IDLE / EP_INVALID
KEY_VALIDATE (Incorrect)	INACTIVE / EP_INACTIVE	EXCHANGING / EP_INVALID	DOWNLOADING/ EP_INVALID	IDLE / EP_INVALID

COMPLETE (Incorrect)	INACTIVE / EP_INACTIVE	EXCHANGING / EP_INVALID	DOWNLOADING/ EP_INVALID	IDLE / EP_INVALID

[0037] In Table 3, the uppermost row shows four states, and the leftmost column shows various messages and their corresponding checked results. For example, supposing that the reset message of RESET is correct after the KDS verifies it, if the current EPi is at downloading state, then it is changed into exchanging state and sends a key exchange message KEY_EXCHANGE to the KDS. In addition to the application of the APS, the present invention can also be applied to any data exchange system in secure demand.

[0038] The above-described embodiments of the present invention are intended to be illustrative only. Numerous alternative embodiments may be devised by persons skilled in the art without departing from the scope of the following claims.